Amendments to the Specification:

Please amend the paragraph (section) beginning on page 9, at line 15 as shown below:

The security processor 102 generally comprises an engine 130 (described in more detail below), an automatic resource (or re-hosting) manager ~~(ARM)~~ security processor (or controller) 132, transport stream encryption/decryption engine configuration logic 134, secure RAM 136, read only memory (ROM) 138, and at least one of a random number generator 150, a hardware multiplier 152, a dynamic feedback arrangement scrambling technique (DFAST) algorithm 154 (i.e., a RAM or ROM that contains the appropriate algorithm), and a hash generation algorithm 156 (e.g., a SHA-1, an MD5, and the like) algorithm (i.e., a RAM or ROM that contains the appropriate algorithm).

Please amend the paragraph (section) beginning on page 10, at line 3 as shown below:

The [[ARM]] processor (or controller) 132 may be coupled to the logic 134, the RAM 136, the firmware 138, the generator 150, the multiplier 152, the DFAST algorithm 154, and a hash generation algorithm 156. The RAM 104 and the flash 106 are generally coupled to the [[ARM]] processor 132. The RAM 104 and the flash 106 may be implemented to provide secure, readily swappable upgrades to the system 100. The controller 132 generally controls the operation of the system 100 in response to at least one (one or more) algorithms (e.g., routines, methods, processes, steps, blocks, procedures, etc. of the predetermined security configuration) that may be stored (i.e., saved, held, etc.) in at least one of the RAM 104, the flash 106, the logic 134, the RAM 136, the ROM 138, the generator 150, the multiplier 152, the DFAST algorithm 154, and the hash 156, as well as internally in connection with the processor 132.

Please amend the paragraph (section) beginning on page 10, at line 15 as shown below:

The [[ARM]] processor (or controller) 132 generally provides for secure downloads, RSA (named after the three inventors - Ron Rivest, Adi Shamir and Leonard

Adleman) key management, multiple key management, digital signatures, and the like, and may include transport stream encryption/decryption logic. The devices (e.g., the logic 134, the RAM 136, the ROM 138, the generator 150, the multiplier 152, the algorithm 154, the hash 156, etc.) may be coupled in parallel. The controller 132 generally couples and controls the appropriate engine or engines 140 and the other devices (e.g., the logic 134, the RAM 136, the ROM 138, the generator 150, the multiplier 152, the algorithm 154, the hash 156, etc.) to meet the design criteria (i.e., the predetermined, desired security configuration) of a particular application.

Please amend the paragraph (section) beginning on page 11, at line 22 as shown below:

In one example, the engine 140a may be implemented as a DES/3-DES stream engine that operates via (i.e., through, using, etc.) a legacy system Cipher Block Chaining (CBC) mode. The legacy CASs use 56-bit DES in CBC mode for the MPEG-2 transport security. The legacy system also uses DFAST scrambling on the DES CBC initialization vector as well as certain DES keys. Triple DES (3-DES) (i.e., application of DES encryption three times using three different keys for a total key bitwidth of 112 or 168 bits) is also used to protect certain structures and the key inside entitlements. The legacy CAS also sends an increment value in the Out Of Band (OOB) channel that is used mathematically with a content key to generate a final DES working key for encrypting or decrypting the MPEG stream packets. The working key is generally changed on a variable frequency as set (i.e., predetermined, selected, etc.) by the headend.

Please amend the paragraph (section) beginning on page 12, at line 4 as shown below:

In one example, the engine 140b may be implemented as a DES/3-DES stream engine that operates via an alternative legacy system Electronic Code Book (ECB) mode. The alternative legacy CAS uses a 56-bit DES in ECB mode for the MPEG-2 transport security. The alternative legacy CAS also uses triple DES (i.e., 112 bit or 168 bit) encryption on the DES keys and to protect entitlements. The alternative legacy CAS also sends a value in the

OOB channel that is used mathematically with the content key to generate a final DES working key for encrypting or decrypting the MPEG stream packets. The working key is generally changed on a variable frequency that is predetermined by the headend.

Please amend the paragraph (section) beginning on page 17, at line 7 as shown below:

The present invention generally provides for generating SHA-1 hash values and for generating Message Digest 5 (MD5) hash values for use in digital signatures (e.g., via the hash generator 156). The present invention generally provides for generation of and verification of digital signatures (e.g., via the [[ARM]] processor 132). Public key signatures for the present invention may be generated and verified using the RSA signature algorithm described in FIPS-PUB 180-1, "Secure Hash Standard".

Please amend the paragraph (section) beginning on page 18, at line 9 as shown below:

The present invention generally provides for the generation of a predetermined number (e.g., in one example, an up to 384-bit, however any appropriate bit-size may be implemented) elliptic curve (EC, i.e., growth) keys and for securely storing the private key for use in digital signatures. The processor of the present invention (e.g., [[ARM]] processor 132) may generate EC-DSA digital signatures securely without exposing (i.e., revealing) the respective private key. The [[ARM]] processor 132 may verify ~~EC-DSA~~ elliptic curve digital signature algorithm (EC-DSA) digital signatures on signed messages and certificates received for authentication.

Amendments to the Drawings:

The attached sheet(s) of drawings includes changes to Fig. 1. This sheet, which includes Fig.1, replaces the original sheet(s) including Fig. 1.

Attachment: Replacement Sheet 1/3
Annotated Sheet 1/3 Showing Changes